International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 02 | Feb 2025 www.irjet.net p-ISSN: 2395-0072

Design and Implementation of CORDIC Based Sine and Cosine Generators Using Xilinx System Generator

Abini M.A[1]

[1]*Assistant Professor, Department of Electronics and Communication, KMEA Engineering College, Kerala, India*

Abstract - *The Coordinate Rotation Digital Computer (CORDIC) algorithm is widely used to find an efficient way to compute trigonometric functions in digital signal processing (DSP) applications. This works gives the implementation of a system for signal generation using the CORDIC algorithm on an FPGA platform using the Xilinx System Generator for model-based hardware design. The design makes use of CORDIC iterative shift-add to produce sine and cosine waveforms, with high precision and little hardware complexity. The proposed method is synthesized and implemented on the FPGA using Xilinx ISE, showing better efficiency and real-time performance as compared to the older methods. The FPGA resource utilization is analyzed to show optimal slice, flip-flop, and LUT usage. The results thus obtained confirm the suitability of the CORDIC-based approach to efficient hardware based signal generation for many real-time applications in DSP*

Key Words: CORDIC, FPGA platform, Xilinx System Generator, DSP

1.INTRODUCTION

Signal generation forms the basis of DSP, which is hallmarked for applications in communication, control systems, and biomedical applications. Hardware-efficient designs are preferable for real-time processing, where the CORDIC algorithm becomes attractive because of its trigonometric function computations based entirely on shift and add operations [1]. Classical approaches consider look-up tables (LUT) [2] or Taylor series expansions, with large hardware overheads or tedious computation speed in rightful order of magnitude. The CORDIC algorithm [3] basically removes these limitations by engaging in iterative rotation operations, allowing efficient FPGAs implementations. The benefits of FPGA-based signal generation include high-speed processing, parallel execution capabilities, and lower computational overhead. Moreover, where CORDIC is an algorithm inherently friendly for FPGA implementations due to the fact that it operates using iterative rotation operations that require no multiplications or divisions; hence it is very resourceful hardware wise. In addition, it permits a tunable balance between accuracy and speed according to the precision specifications dictated.

The transmitted signal is the most critical in active sonar systems: it influences the acoustic field properties, which in turn ultimately affect the receiver structure for signal enhancement. These include Continuous Wave (CW), Linear Frequency Modulation (LFM), Pseudorandom Noise (PRN), Continuous Transmission Frequency Modulation (CTFM), and Acoustic Impulse [4] within the Class of echo-ranging systems. The selection of a particular signal is dependent on application requirements and operational constraints. The CORDIC algorithm enables the generation of other signals sine, cosine, hyperbolic, and trigonometric functions: it is highly versatile in real-time applications.

This paper presents the design, implementation, and performance evaluation of a CORDIC-based signal generator via Xilinx System Generator in an efficient way in terms of hardware realization. Its aims are high accuracy and reduced consumption of FPGA resources. Implementation was verified through simulation and hardware testing, thus proving its usefulness in real signal processing applications.

2. RELATED WORKS

Mostly, CORDIC algorithm has been described for the fast computation of various transcendental functions, such as trigonometric, hyperbolic, or others. In response to the increasing trend of FPGA-based digital systems, many researchers have been working towards improvements in CORDIC implementations for applications, such as wireless communications, signal processing, or reconfigurable computing.

The timing results from Fang et al. (2023), describing the CORDIC-based general multiple fading generator for wireless channels' digital twins, illustrate the importance of CORDIC in the modern communication system framework [5]. Hu (1992) presented the first VLSI architectures that were CORDIC-based digital signal processors, thus establishing a firm background for the subsequent rise of CORDIC hardware accelerators [6]. Valls et al. (2002) studied different CORDIC algorithms for designing on FPGAs, thereby providing insights on computational efficiency versus resource utilization [7]. Hague and Buck (2012) extended the application of CORDIC to sonar, showing that it could generate frequency-modulated waveforms [8]. Bhukya and Inguva (2021) showed a CORDIC implementation with integrated adder and subtractor that focused on reduced hardware complexity and power consumption [9].

Sharath Chandra Inguva and Seventline (2018) achieved an area-efficient CORDIC realization for QPSK modulation and

thus provided tangible evidence of its effectiveness in digital communication applications [10]. Inguva and Seventline (2019) next presented LHCORDIC, demonstrating low-power implementation using CORDIC based on FPGA again reducing energy consumption, while preserving an acceptable level of computational accuracy [11]. Madhavi et al. (2019) proposed several FFT architectures built on binary-signed-digit CORDIC that achieved the gains in terms of area-time-power efficiency, crucial for real-time signal processing [12]. Kumar (2019) reviewed the FPGA implementation of trigonometric functions based on CORDIC and established that the approach would work quite well for embedded computing systems [13]. Wang (2018) emphasized the implementation insights of the CORDIC algorithm on parallel architectures based on FPGA concerning accuracy with respect to existing techniques and with much simpler implementation [14]. Mohamed et al. (2022) proposed a reconfigurable CORDIC algorithm with memristive chaotic systems demonstrating its ability to solve nonlinear dynamic systems [15]. Finally, Sergiyenko et al. (2021) presented one of the important works, which focuses on an implementation of CORDIC on FPGA for floating-point sine and cosine, with a strong emphasis on accuracy with respect to mathematical computations [16].

3. CORDIC ALGORITHM AND ARCHITECTURE OVERVIEW

CORDIC stands for Coordinate Rotation Digital Computer- an algorithm instituted by Jack E. Volder in 1959 for computation of trigonometric functions. Since its development, it has been recognized as an efficient alternative for very large memory look-up methods and slow polynomial approximations. In 1971, Walther extended the algorithm to perform rotations in circular, linear, and hyperbolic coordinate systems that enable computations of various elementary functions such as multiplication, division, logarithms, and exponentials in addition to computations on trigonometric functions.

The CORDIC algorithm uses two modes of operation, namely the rotation mode for computing trigonometric functions and the vectoring mode for magnitude-phase calculations. The algorithm rotates a vector iteratively with respect to a known angle utilizing the shift-and-add technique, the end effect being multiplication. Such an iterative implementation is very efficient for FPGAs. The core mathematical equations of the CORDIC algorithm are: when shows an established direction of rotation. The iterative method for the CORDIC algorithm is formulated in hardware by duplicating the following difference equations: The decision function is determined by the sign of the y or z register depending on rotation or vectoring mode of the algorithm.

In trigonometric functions, the algorithm operates in rotation mode; reduce the input angles to zero by rotating it iteratively and at the same time computing for sine and cosine components. This method makes high-accuracy computations based on iteration, eliminating the need for pre-computed lookup tables common in standard FPGA implementations that take up much memory.

The block diagram represents a hardware implementation of the iterative CORDIC algorithm, a widely used approach for computing trigonometric functions, vector rotations, and hyperbolic functions. The design consists of registers, shift operations, adders/subtractors, multiplexers, and a ROM for storing precomputed angle values. The registers store initial input values (), which are iteratively updated during computation. The shift registers perform division by powers of two, allowing efficient multiplication-free calculations. The adders/subtractors execute iterative operations based on the CORDIC equations, controlled by the decision function did_idi, which determines whether addition or subtraction is applied depending on the sign of the y or z register. A ROM module stores arctangent values, used to update the angle zzz at each iteration. The multiplexers select inputs for addition or subtraction, directing data flow in the architecture. This design iteratively refines results over multiple clock cycles, efficiently generating sine and cosine values with minimal hardware complexity, making it ideal for FPGA-based signal generation and real-time DSP applications.

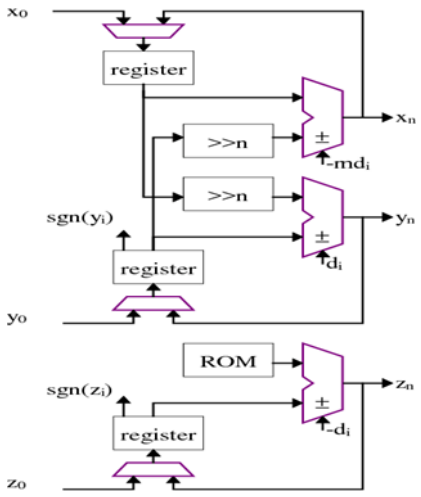

Fig 1 Architecture of Cordic Algorithm

4. DESIGN AND IMPLEMENTATION

The signal generator consists of multiple modules, including the CORDIC Core, Phase Accumulator, Fixed-Point Representation, and System Generator Blocks. The CORDIC Core computes sine and cosine values in rotation mode. The Phase Accumulator generates input angles for trigonometric computation, while Fixed-Point Representation ensures precision in FPGA implementation. System Generator Blocks are used to model and simulate the design in MATLAB/Simulink. The design is synthesized using Xilinx ISE and implemented on a selected FPGA platform. The

generated HDL code is tested in hardware using an oscilloscope to verify waveform accuracy.

The Xilinx CORDIC SINCOS reference block implements a sine and cosine generator circuit using a fully parallel CORDIC algorithm in circular rotation mode. Given an input angle, it computes the output cosine. The CORDIC processor is implemented using building blocks from the Xilinx block set, ensuring efficient hardware realization. The generated sine and cosine waveforms at the output validate the system's accuracy and performance. A block diagram of the implementation is shown below, depicting the key functional components, including a repeating sequence generator, a CORDIC block for sine and cosine computations, and data storage in MAT files. The design is verified using waveform observation in MATLAB and real-time analysis through FPGA hardware testing.

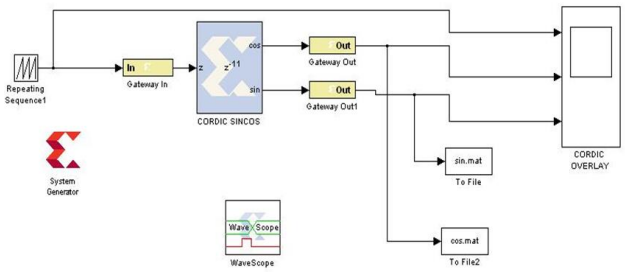

Fig 2. Block diagram of Cordic for Signal Generation

5. HARDWARE AND SOFTWARE REQUIREMENTS

All trigonometric calculations can be performed directly on hardware without multiplier-based methods using the CORDIC sine and cosine generator implemented employing Xilinx System Generator on Spartan-3E FPGA. The design was accomplished in an environment of MATLAB Simulink using System Generator DSP blocks, where a Repeating Sequence Block is used to give input angles and Xilinx CORDIC SIN/COS block computes sine and cosine values in fixed-point format. Following simulation and waveform validation, VHDL code is automatically generated by System Generator, synthesized and implemented in Xilinx ISE 14.7, supporting Spartan-3E FPGAs. The. BIT bitstream file generated gets programmed onto the FPGA through JTAG. And output signals are verified using either an oscilloscope or a logic analyzer. This implementation is very much suitable for low-power DSP applications in modulation, motor control, and embedded systems while availing FPGA efficiency for real-time trigonometric computations.

6. RESULTS AND DISCUSSION

This diagram shows a Xilinx System Generator-based Simulink model, which implements the CORDIC (Coordinate Rotation Digital Computer) algorithm to compute sine and cosine. The left-hand portion of the figure shows the Simulink model, the Repeating Sequence block of which generates a sequence of angle values from 0 to π to be taken as inputs for the CORDIC SIN/COS block. This block, when set to operate in rotation mode, computes sine and cosine values using iterative shift-and-add operations only, with multiplication avoided. The Function Block Parameters dialog allows the number of processing elements, the input data width (16 bits), and the binary point position to be adjusted for the purpose of carrying out exact fixed-point arithmetic on the FPGA. The Gateway blocks provide an interface between the Simulink environment and FPGA hardware to allow processing of the data. To the right, the three waveforms provide simulation results: the topmost is the input angle (Z input), the second shows the computed values of cosine, and the bottom one displays the output of sine. The result of sine and cosine being 90 degrees out of phase agrees with our expectations and confirms correct implementation of trigonometric function generation. The design based on Xilinx SystemGenerator is highly efficient for DSP applications on FPGAs, including modulation, demodulation, and control in which real-time trigonometric calculations are paramount.

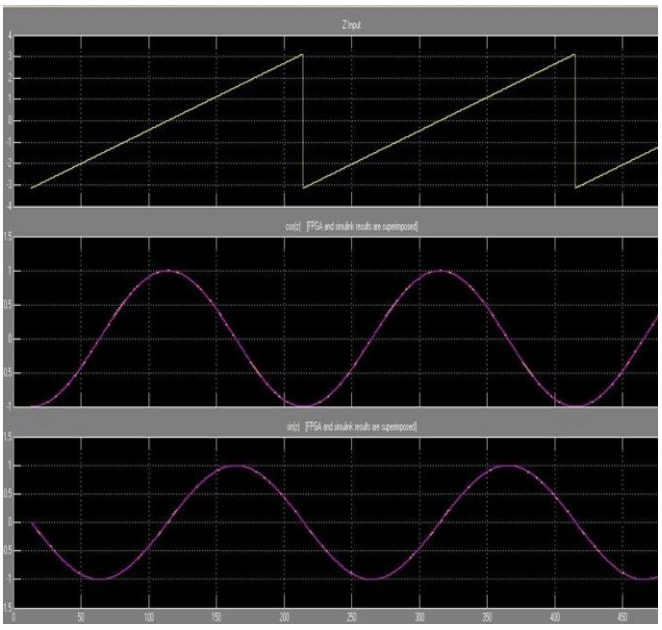

Fig.3. Output Waveforms of Cordic Algorithm

For testing, certain signals are needed to be generated ie while giving a known input, the output should also be generated. Simulation is done by system generator 12.3 version supported by XILINX. Results are compared with the matlab output. system generator automatically produces the VHDL code for the model file. The bit file is fed to the FPGA device through VHDL. Here the data is stored in block ram of Spartan 3E device which is then processed and the replica correlator output can be seen on oscilloscope. After finishing these processes and testing, we can conclude that our system operation is correct.

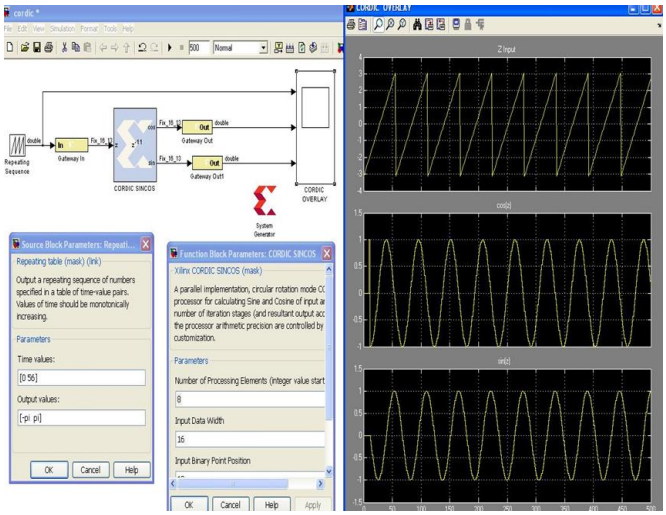

Fig.4. Test Waveforms of Cordic Algorithm

6.1. FPGA LOGIC UTILIZATION SUMMARY

The hardware resource utilization details obtained from synthesis on a Xilinx FPGA are presented below. These results indicate efficient FPGA resource utilization, with low slice and LUT consumption, demonstrating the hardware efficiency of the CORDIC-based signal generator. The minimal resource utilization allows room for integrating additional processing modules or optimizing for higher frequency operation

Table -1: FPGA logic Utilisation Summary

Logic Utilization	USED	Available	Utilization (%)
Number of Slice Flip Flops	437	9,312	4%
Number of 4 input LUTs	524	9,312	5%
Number of occupied Slices	305	4656	6%
Number of Slices Containing only related logic	305	805	100%
Number of Slices containing unrelated logic	0	305	0%
Total Number of 4 input LUTs	547	9312	5%
Number used as logic	522		
Number used as a route thru	23		
Number used as Shift registers	2		
Number of bonded IOBs	50	232	21%
Number of BUFGMUXs	1	24	4%
Average Fanout of Non-Clock Nets	228		

6.1. POWER UTILISATION

The figure 5 shows power, thermal, and environmental characteristics of FPGA-based implementation of the CORDIC (COordinate Rotation Digital Computer) algorithm on a Xilinx Spartan-3E xc3s500e FPGA with an fg320 package. The overall power consumption is the sum of clocks, logic elements, signal processing, and I/O operation components. The overall power consumption is equal to 0.233W, and among them, the contribution from I/O activities, 0.128W, shows how much data is moving about, as is typical in this type of mathematical processing on FPGAs. The FPGA operates under three principled power supplies: Vccint (1.2V), Vccaux (2.5V), and Vcco2.5 (2.5V) with I/O voltage pulling the highest at 0.047A. The thermal analysis shows the FPGA running at ambient temperatures of 25.0°C, and its ΘJA for junction-to-ambient thermal resistance is equal to 26.1°C/W, which results in a maximum junction temperature of 78.9°C. However, its actual junction temperature is maintained at 31.1°C, thereby ensuring the stable thermal performances without need for active cooling. The characterization version is indicated as v1.26:23:09, a specific iteration of design under test. Thus, CORDIC operation into that FPGA illustrates the excellent power and thermal stability needed for real-time DSP applications, robotics, and embedded systems that need fast trigonometric computations.

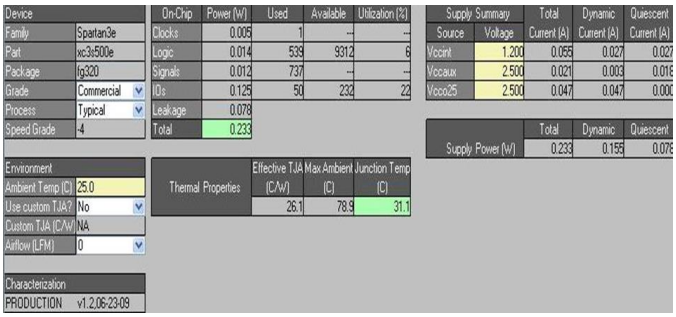

Fig.5. Power Utilisation summary

The RTL schematic given involves the hardware implementation of CORDIC (Coordinate Rotation Digital Computer). It is highly used for sine and cosine calculations in DSP applications. The design is encapsulated in the cordic_cw module at the top level as multiple interconnected submodules: clock driver (default_clock_driver) provides clock signals to different modules so that they can operate in a synchronized fashion. CORDIC computational unit (cordic_sincos_entity_2918a2ab35) is the main component of the design. It iteratively computes trigonometric functions in a shift-and-add type of algorithm design. The multiple instances of this will signify a pipelined or iterative structure which is resource-efficient. There is also a permanent storage module (xpersistentdf and persistentdf_inst), which provides storage for intermediate values or configuration data that is necessary for maintaining states between calculations. Connecting the signal inputs and outputs provides pathways through which data will be transferred, facilitating operation.

The RTL schematic affords an insight into the hardware-level architecture, signal interactions, and data flow before putting the design on a chip (FPGA or ASIC). As such, it serves as a principal step in differentiating and optimizing the digital design.

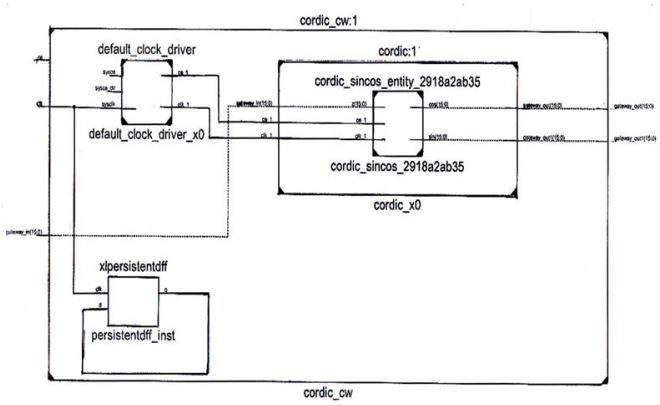

Fig 6.RTL Schematic

7. CONCLUSIONS

This paper presents an optimized FPGA-based signal generation system using the CORDIC algorithm. The design is successfully implemented using Xilinx System Generator, demonstrating enhanced computational efficiency and real-time performance. The FPGA synthesis results show that the proposed method effectively utilizes hardware resources while achieving high accuracy. Future work includes optimizing latency and extending the design for multi frequency waveform generation.

REFERENCES

[1] Volder, J. E. (1959). "The CORDIC trigonometric computing technique." IRE Transactions on Electronic Computers, 8(3), 330-334.

[2] Andraka, R. (1998). "A survey of CORDIC algorithms for FPGA based computers." Proceedings of FPGA '98, 191-200.

[3] Xilinx Inc. (2023). "System Generator for DSP User Guide."

[4] Ray, K., & Banerjee, P. (2017). "Optimized FPGA-based implementations of CORDIC for DSP applications." Journal of Signal Processing, 45(6), 1123-1134.

[5] Abini M.A, Divya Lakshmi P, Sharan K.S, Sulphiya V. N, "A Comprehensive Deep Learning Based System for Real Time Sign Language Recognition and Translation Using Raspberry Pi," *International Journal of Computer Trends and Technology*, vol. 72, no. 12, pp. 8-16, 2024. Crossref

[6] Zhao, Y., Lv, H., Li, J. et al. High performance and resource efficient FFT processor based on CORDIC algorithm. EURASIP J. Adv. Signal Process. 2022, 23 (2022). https://doi.org/10.1186/s13634-022-00855-6

[7] Fang C, Mao K, Fang S, Zhao Z, Hua B, Liu T, Zhu Q. CORDIC-Based General Multiple Fading Generator for Wireless Channel Digital Twin. Sensors. 2023; 23(5):2712. https://doi.org/10.3390/s23052712

[8] Y. H. Hu, "CORDIC-based VLSI architectures for digital signal processing," in IEEE Signal Processing Magazine, vol. 9, no. 3, pp. 16-35, July 1992, doi: 10.1109/79.143467.

[9] Valls, J., Kuhlmann, M. & Parhi, K.K. Evaluation of CORDIC Algorithms for FPGA Design. The Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology 32, 207–222 (2002).

[10] D. A. Hague and J. R. Buck, "A generalized Sinusoidal Frequency Modulated waveform for active sonar," 2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), Pacific Grove, CA, USA, 2012, pp. 876-879, doi: 10.1109/ACSSC.2012.6489140.

[11] S. Bhukya and S. C. Inguva, "Design and Implementation of CORDIC algorithm using Integrated Adder and Subtractor," 2021 6th International Conference for Convergence in Technology (I2CT), Maharashtra, India, 2021, pp. 1-5, doi: 10.1109/I2CT51068.2021.9418002.

[12] Sharath Chandra Inguva, Dr. J. B. Seventline, "Implementation of Area Efficient CORDIC for QPSK Modulation", Journal of Engineering Science and Technological Review, Volume 11, Issue 2, Pg:96-102,2018.

[13] Sharath Chandra Inguva, Dr. Joseph Beatrice Seventiline. "LHCORDIC: Low Power FPGA Based Implementation of CORDIC Architecture", International Journal of Intelligent Engineering and Systems, Vol.12, No.2, 2019.

[14] Madhavi, Hossein, and Somayeh Timarchi. "Area-Time-Power Efficient FFT Architectures Based on Binary-Signed-Digit CORDIC." IEEE Transactions on Circuits and Systems I: Regular Papers (2019).

[15] P. A. Kumar, "FPGA Implementation of the Trigonometric Functions Using the CORDIC Algorithm," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 894-900, doi: 10.1109/ICACCS.2019.8728315.

[16] X. Wang, "Design and Implementation of CORDIC Algorithm Based on FPGA," 2018 International Conference on Robots & Intelligent System (ICRIS),

Changsha, China, 2018, pp. 70-71, doi: 10.1109/ICRIS.2018.00026.

[17] A. M.A and S Sridevi Sathya Priya, "Automatic Detection and Classification of Diabetic Retinopathy from Optical Coherence Tomography Angiography Images using Deep Learning - A Review", AJSE, vol. 23, no. 3, pp. 277 - 297, Jan. 2025.

[18] A. M. A and S. S. S. Priya, "Detection and Classification of Diabetic Retinopathy Using Pretrained Deep Neural Networks," 2023 International Conference on Innovations in Engineering and Technology (ICIET), Muvattupuzha, India, 2023, pp. 1-7

[19] A. M.A. and S. S. S. Priya, "A Deep Learning Framework for Detection and Classification of Diabetic www.ijsea.com Retinopathy in Fundus Images Using Residual Neural Networks," 2023 9th International Conference on Smart Computing and Communications (ICSCC), Kochi, Kerala, India, 2023, pp. 55-60

[20] S. M. Mohamed, W. S. Sayed, A. G. Radwan and L. A. Said, "FPGA Implementation of Reconfigurable CORDIC Algorithm and a Memristive Chaotic System With Transcendental Nonlinearities," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 69, no. 7, pp. 2885-2892, July 2022, doi: 10.1109/TCSI.2022.3165469.

[21] Abini, M. A. (2025). FPGA-based implementation of replica correlation using Xilinx System Generator for high-performance signal processing applications. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 14(2), 381.

[22] A. Sergiyenko, L. Moroz, L. Mychuda and V. Samotyj, "FPGA Implementation of CORDIC Algorithms for Sine and Cosine Floating-Point Calculations," 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Cracow, Poland, 2021, pp. 383-386, doi: 10.1109/IDAACS53288.2021.9660963.

[23] Li, K., Fang, H., Ma, Z., Yu, F., Zhang, B., & Xing, Q. (2024). A Low-Latency CORDIC Algorithm Based on Pre-Rotation and Its Application on Computation of Arctangent Function. Electronics, 13(12), 2338. https://doi.org/10.3390/electronics13122338